

## CLAIMS

1. A method for converting a source program into one or more tokens, comprising:  
obtaining one or more entries;  
5 analyzing said source program; and  
generating said tokens from said source program, wherein said entries may be  
used to generate a subset of said tokens.

2. The method of claim 1 wherein said entries are comprise a language descriptor  
10 and a token value.

3. The method of claim 2 wherein the analyzing comprises:  
obtaining a lexeme from said source program; and  
determining if said lexeme matches one of said language descriptors.

4. The method of claim 3 wherein the analyzing further comprises:  
obtaining said token value if said lexeme matches one of said language  
descriptors.

5. The method of claim 4 wherein the analyzing further comprises:  
obtaining a next lexeme from said source program.

6. The method of claim 5 wherein the generating comprises:  
outputting said token value in response to a request from a host program

7. The method of claim 6 wherein said language descriptor is a reserved word.

8. The method of claim 6 wherein said language descriptor is an operator.

5 9. The method of claim 1 wherein the obtaining further comprises:  
entering said token entries into a token dictionary.

10. A computer program product comprising:

a computer usable medium having computer readable program code embodied  
10 therein configured to convert source program into one or more tokens, said computer  
program product comprising:

computer readable code configured to cause a computer to obtain one or more  
entries;

computer readable code configured to cause a computer to analyze said source  
15 program; and

computer readable code configured to cause a computer to generate said tokens  
from said source program, wherein said entries may be used to generate a subset of said  
tokens.

20 11. The computer program product of claim 10 wherein said entries comprise a  
language descriptor and a token value.

12. The computer program product of claim 11 wherein said computer code  
configured to cause a computer to analyze the source program comprises:

computer readable code configured to cause a computer to obtain a lexeme from said source program; and

computer readable code configured to cause a computer to determine if said lexeme matches one of said language descriptors.

5

13. The computer program product of claim 12 wherein said computer code configured to cause a computer to analyze said source program further comprises:

computer readable code configured to cause a computer to obtain said token value if said lexeme matches one of said language descriptors.

10

14. The computer program product of claim 13 wherein said computer code configured to cause a computer to analyze said source program further comprises:

computer readable code configured to cause a computer to obtain a next lexeme from said source program.

15

15. The computer program product of claim 14 wherein said computer code configured to cause a computer to generate tokens comprises:

computer readable code configured to cause a computer to output said token value in response to a request from a host program

20

16. The computer program product of claim 15 wherein said language descriptor is a reserved word.

17. The computer program product of claim 15 wherein said language descriptor is an operator.

25

18. The computer program product of claim 10 wherein said computer code configured to cause a computer to obtain one or more entries further comprises:

computer readable code configured to cause a computer to enter said token entries  
5 into a token dictionary.

19. A lexical analyzer, comprising:

one or more entries configured to be obtained;

a source program analyzer;

10 one or more tokens configured to be generated from said source program analyzer, wherein said entries may be used to generate a subset of said tokens.

20. The lexical analyzer of claim 19, wherein said entries comprise a language descriptor and a token value.

15 21. The lexical analyzer of claim 20, wherein said source program analyzer comprises:

an source program interface, wherein said interface obtains a lexeme from said a source program; and

20 a lexeme comparator, wherein said comparator determines whether said lexeme matches one of said language descriptors.

22. The lexical analyzer of claim 21, wherein said source program analyzer further comprises:

a token output interface, wherein said interface generates said token if said lexeme matches one of said language descriptors.

23. The lexical analyzer of claim 22, wherein said source program interface further  
5 comprises:

a source program manager, wherein said manager obtains a next lexeme from said source program.

24. The lexical analyzer of claim 23, wherein said output interface comprises:

10 a host program event handler, wherein said event handler causes said output interface to generate said token value in response to a request from the host program.

25. The lexical analyzer of claim 24, wherein said language descriptor is a reserved word.

15 26. The lexical analyzer of claim 24, wherein said language descriptor is an operator.

27. The lexical analyzer of claim 19, further comprising:

a token dictionary, wherein said entries comprise dictionary entries.